CLAIMS:

1. A system for generating a true random number comprising:

a microprocessor operating at a first frequency,

at least one counter for generating bits;

at least one shifter for scrambling bits;

at least one first oscillator for cooperating with said at least one counter; and

at least one second oscillator for cooperating with said at least one shifter,

wherein said oscillators provide a frequency perturbation based on digital input signals initialized via said microprocessor.

2. The system of claim 1, wherein said counter has an initialization register for receiving an initialization bit value.

3. The system of claim 2, wherein said initialization bit value is at a trailing edge of an initialization write of said microprocessor.

4. The system of claim 3, wherein said at least one first oscillator is a ring oscillator having a first odd number of stages.

5. The system of claim 4, wherein said at least one first ring oscillator is cooperates with said at least one counter to provide a second frequency.

6. The system of claim 5, wherein said at least one shifter is a barrel shifter being continuously spun by said at least one second oscillator at a third frequency.

7. The system of claim 6, wherein said at least one second oscillator is a ring oscillator having a second odd number of stages differing from said first odd number of stages by at least two stages.

8. The system of claim 7, wherein said third frequency is asynchronous to said second frequency.

9. The system of claim 8, wherein said third frequency is asynchronous to said first frequency.

10. The system of claim 8, wherein said third frequency is asynchronous to and faster than said first frequency.

11. The system of claim 9, wherein said counter is timed or clocked at said second frequency with said second frequency being asynchronous to said third frequency.

12. The system of claim 11, wherein said second frequency is asynchronous to said first frequency.

13. The system of claim 12, wherein when said microprocessor reads a random number, said barrel shifter inputs a current counter bit value and shifts said bit value by a current barrel shift count.

14. A method for providing a true random number generator comprising the steps of:

(a) providing a microprocessor operating at a first frequency;

(b) providing at least one counter;

(c) providing at least one first oscillator to clock said at least one counter at a second frequency;

(d) providing at least one shifter; and

(e) providing at least one second oscillator for continuously spinning said at least one shifter at a third frequency.

15. The method of claim 13, wherein said at least one first oscillator has a first odd number of stages and said second oscillator has a second odd number of stages differing from said first odd number of stages by at least two stages.

16. The method of claim 13, wherein said first frequency, said second frequency and said third frequency are each asynchronous to each other.

17. The method of claim 15, wherein when said microprocessor reads a random number, said shifter inputs a current counter bit value and shifts said bit value by a current shift count.

18. A method for generating a true random number comprising the steps of:

(a) providing a microprocessor operating at a first frequency, at least one counter for generating bits, at least one shifter for scrambling bits, a first and second oscillator for cooperating with said counter and said shifter, respectively;

(b) initializing said counter by a write of said microprocessor to an initialization register of said at least one counter;

(c) clocking said at least one counter via said first oscillator at a second frequency;

(d) continuously spinning said at least one shifter via said second oscillator at a third frequency;

(e) inputting a current counter bit value, at a time when said microprocessor reads a random bit number, and shifting said current bit value by a current shift count; and

(f) returning said shifted bit value to said microprocessor to achieve a non-predictable pattern of bit numbers.

19. The method of claim 17, wherein said at least one first oscillator has a first odd number of stages and said second oscillator has a second odd number of stages differing from said first odd number of stages by at least two stages.

20. The method of claim 17, wherein said first frequency, said second frequency and said third frequency are each asynchronous to each other.

21. The method of claim 19, wherein when said microprocessor reads a random number, said shifter inputs a current counter bit value and shifts said bit value by a current shift count.